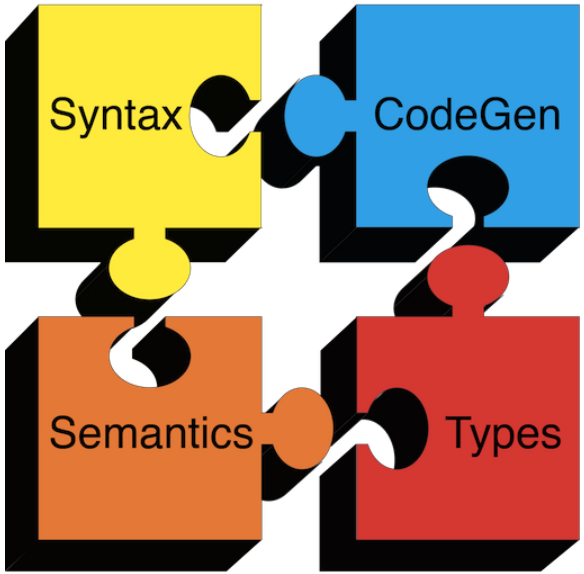


Compilers

Derivations

Derivations

A *derivation* is a sequence of productions

$$S \rightarrow \ldots \rightarrow \ldots \rightarrow \ldots \rightarrow \ldots \rightarrow \ldots$$

A derivation can be drawn as a tree

- Start symbol is the tree's root
- For a production $X \rightarrow Y_1 \ldots Y_n$ add children $Y_1 \ldots Y_n$ to node X

Derivations

- Grammar

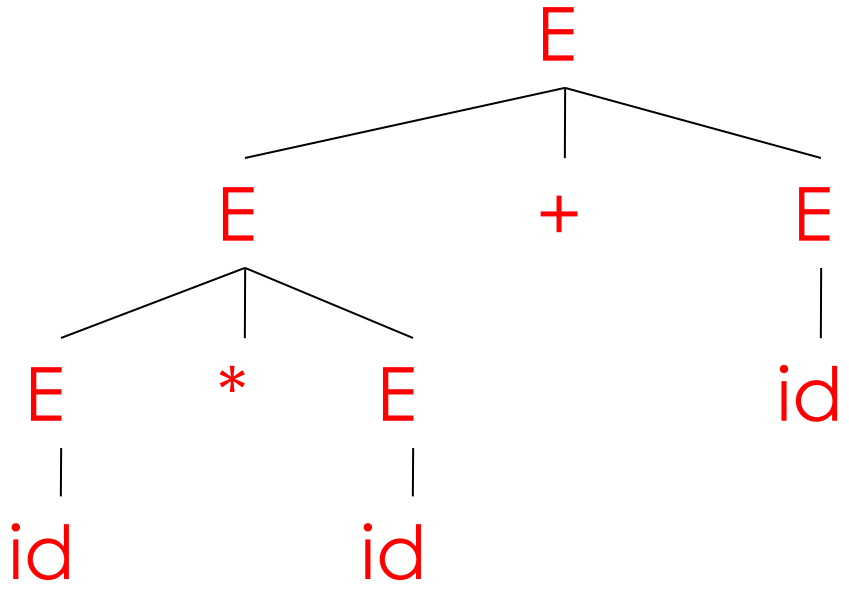
$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

- String

$$id * id + id$$

Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E*E+E \\
\rightarrow\ & id*E+E \\
\rightarrow\ & id*id+E \\
\rightarrow\ & id*id+id
\end{aligned}
$$



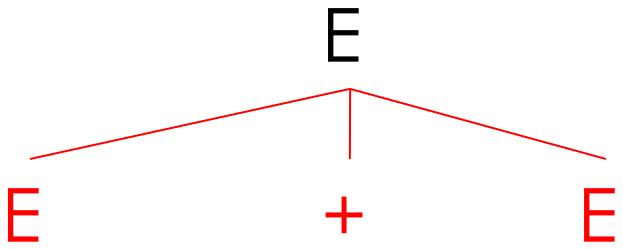
Derivations

E

E

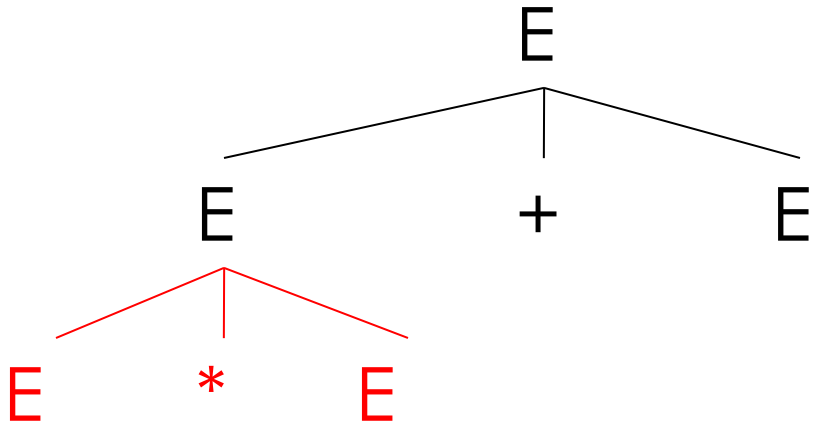
Derivations

$$\begin{array}{ll} & \mathbf{E} \\ \rightarrow & \mathbf{E+E} \end{array}$$



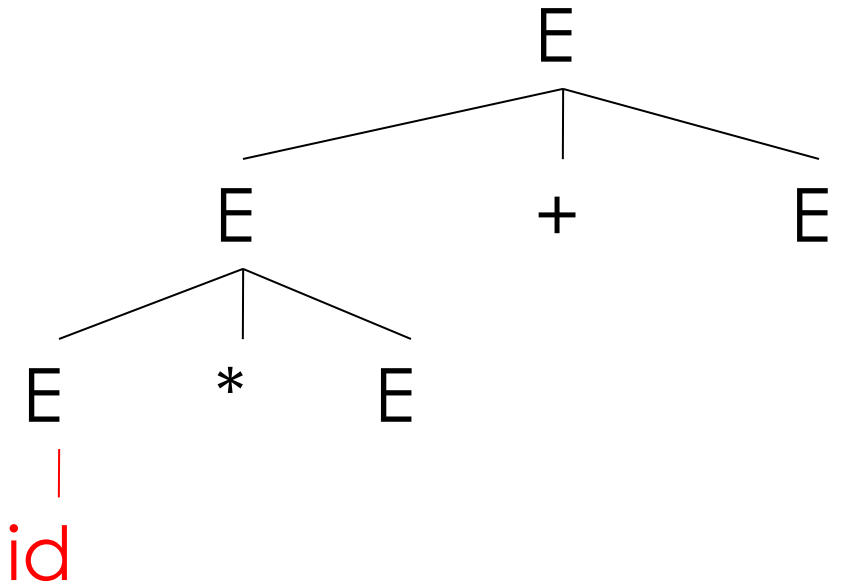
Derivations

$$
\begin{aligned}
& \mathbf{E} \\
\rightarrow\ & \mathbf{E}+\mathbf{E} \\
\rightarrow\ & \mathbf{E}*\mathbf{E}+\mathbf{E}
\end{aligned}
$$



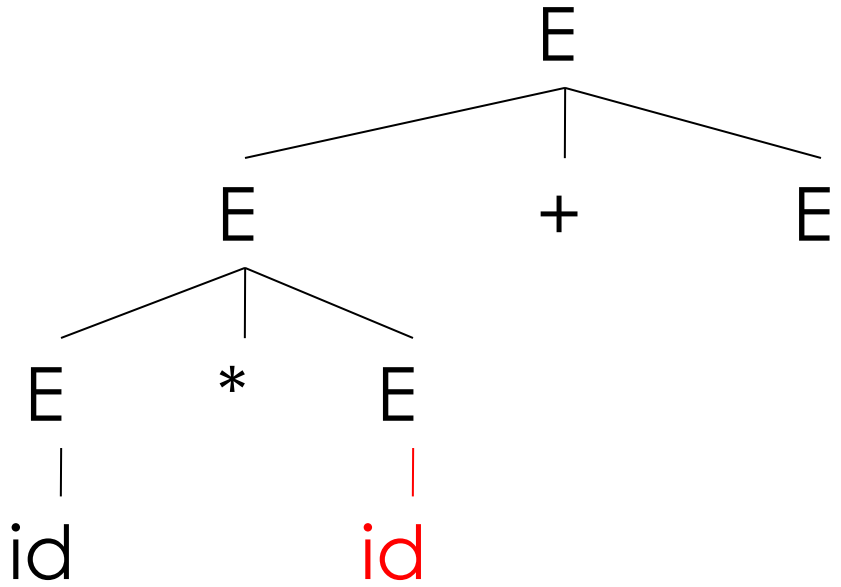
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E + E \\
\rightarrow\ & E * E + E \\
\rightarrow\ & id * E + E
\end{aligned}
$$



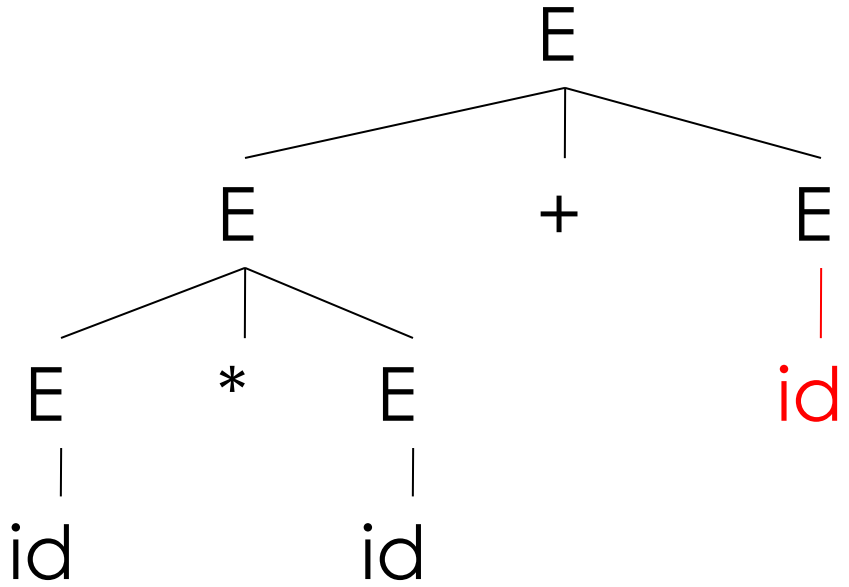
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E * E+E \\
\rightarrow\ & id * E + E \\
\rightarrow\ & id * id + E
\end{aligned}
$$



Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E*E+E \\
\rightarrow\ & id*E+E \\
\rightarrow\ & id*id+E \\
\rightarrow\ & id*id+id
\end{aligned}
$$



Derivations

- A parse tree has
  - Terminals at the leaves
  - Non-terminals at the interior nodes
- An in-order traversal of the leaves is the original input
- The parse tree shows the association of operations, the input string does not

Derivations

- The example is a *left-most* derivation
  - At each step, replace the left-most non-terminal
- There is an equivalent notion of a *right-most* derivation

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E+id \\
\rightarrow\ & E*E+id \\
\rightarrow\ & E*id+id \\
\rightarrow\ & id*id+id
\end{aligned}
$$

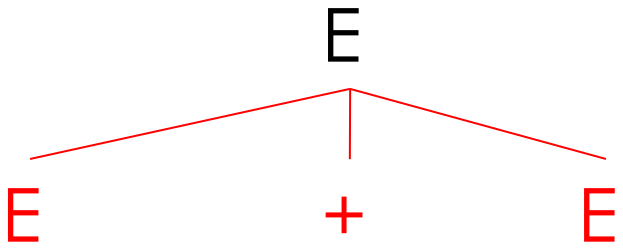
Derivations

E

E

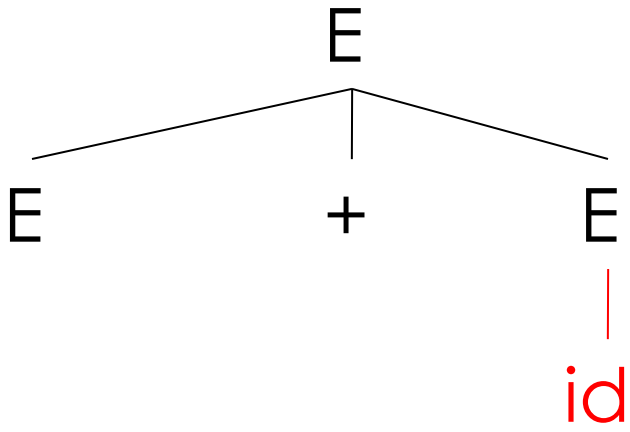
Derivations

$$\begin{array}{ll} & \mathbf{E} \\ \rightarrow & \mathbf{E+E} \end{array}$$



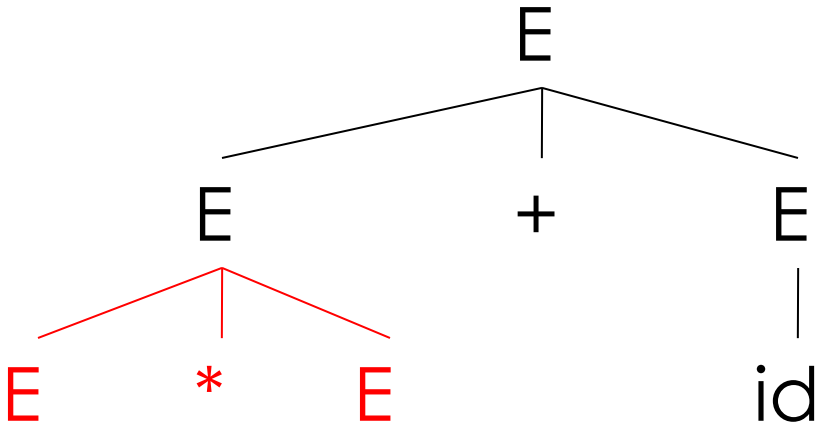
Derivations

$$
\begin{aligned}
& \mathbf{E} \\
\rightarrow\ & \mathbf{E+E} \\
\rightarrow\ & \mathbf{E+id}
\end{aligned}
$$



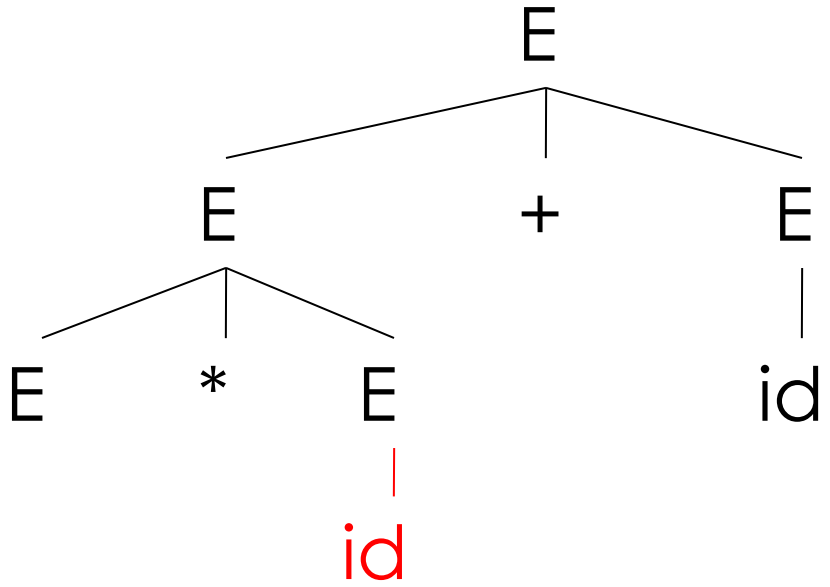
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E+id \\
\rightarrow\ & E*E+id
\end{aligned}
$$



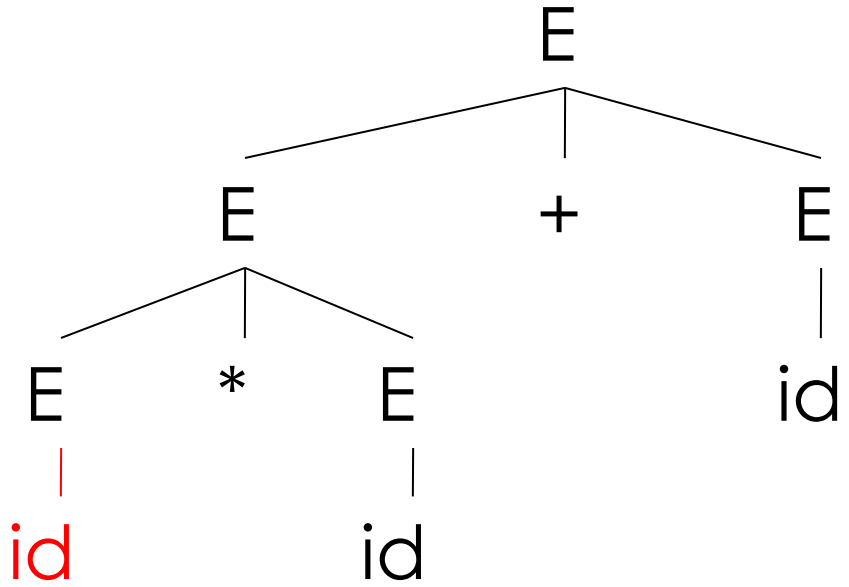
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E+id \\
\rightarrow\ & E*E+id \\
\rightarrow\ & E*id+id
\end{aligned}
$$



Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E+id \\
\rightarrow\ & E*E+id \\
\rightarrow\ & E*id+id \\
\rightarrow\ & id*id+id
\end{aligned}
$$



Derivations

Note that right-most and left-most derivations have the same parse tree

Derivations

Which of the following is a valid derivation of the given grammar?

$S \rightarrow aXa$

$X \rightarrow \varepsilon \mid bY$

$Y \rightarrow \varepsilon \mid cXc \mid d$

- S
  aXa
  abYa
  acXca
  acca

- S
  aa

- S
  aXa
  abYa
  abcXca
  abcbYca
  abcbdca

- S
  aXa
  abYa
  abcXcda
  abccda

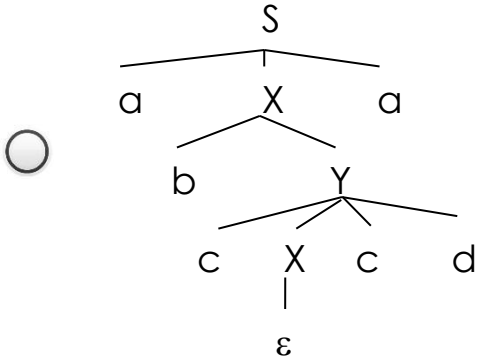
Derivations

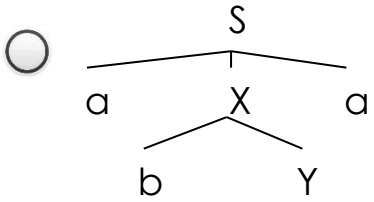
Which of the following is a valid parse tree for the given grammar?

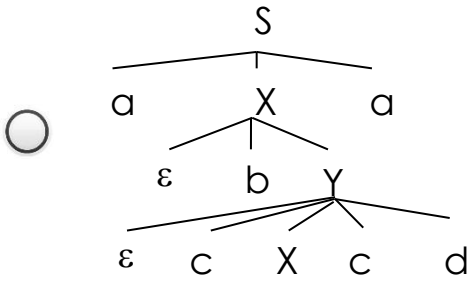
$S \rightarrow aXa$

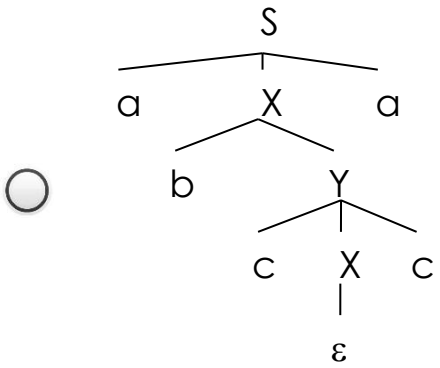
$X \rightarrow \varepsilon \mid bY$

$Y \rightarrow \varepsilon \mid cXc \mid d$









Derivations

- We are not just interested in whether $s \in L(G)$
  - We need a parse tree for s
- A derivation defines a parse tree
  - But one parse tree may have many derivations
- Left-most and right-most derivations are important in parser implementation